ATTORNEY DOCKET NO.
SUNMP232

PATENT APPLICATION

EFFICIENT UTILIZATION OF SHARED BUFFER MEMORY AND METHOD FOR OPERATING THE SAME

INVENTORS: (1) Whay Sing Lee
37171 Sycamore St., Apt. 1438
Newark, CA 94560
Citizen of Malaysia

(2) Walter Nixon
3317 Hogarth Place
Fremont, CA 94555
Citizen of the United States of America

(3) Fay Chong, Jr.
10019 Oakleaf Place
Cupertino, CA 95014
Citizen of the United States of America

ASSIGNEE: Sun Microsystems, Inc.
4150 Network Circle
Santa Clara CA 95054

MARTINE & PENILLA, LLP
710 LAKEWAY DRIVE, SUITE 170
SUNNYVALE, CA 94085

Efficient Utilization of Shared Buffer Memory and Method for Operating the Same

by Inventors

Whay Sing Lee

Walter Nixon

Fay Chong, Jr.

CROSS REFERENCE TO RELATED APPLICATIONS

[1] This application is related to U.S. Patent Application No. __________ (Attorney Docket No. SUNMP233), filed on even date herewith, and entitled "Bandwidth Matching Device and Method for Operating the Same," and U.S. Patent Application No. __________ (Attorney Docket No. SUNMP234), filed on even date herewith, and entitled "Time Slicing Device for Shared Resources and Method for Operating the Same." The disclosures of both of these related applications are incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[2] The present invention relates generally to a digital memory.

2. Description of the Related Art

[3] In modern electronic devices, switches are used to connect a number of senders to a number of receivers, such that any of the senders can deliver a data stream (or packet) to any of the receivers. To accommodate variability in communication between the senders and receivers, the switches may use packet buffer memories to store packets received from the senders, which cannot immediately be transmitted to the intended receivers.

[4] One conventional packet buffer memory configuration uses an input buffer connected to each of a number of input ports, wherein a different input buffer is connected to a different one of the number of input ports. Each of the different input

buffers associated with the number of input ports is connected to a switching substrate (e.g., crossbar switch or shared bus) which is connected to a number of output ports. In this configuration, if there are two packets stored in the same input buffer, but destined for different output ports, only one packet can make progress even if both output ports are available. This situation exists because the two packets share a common connectivity into the switching substrate. Also in this configuration, the input buffer associated with the input port that is receiving the packet must have sufficient space available to receive the entire packet. The packet cannot be stored in another input buffer associated with another input port.

[5] Another conventional packet buffer memory configuration uses an output buffer connected to each of a number of output ports, wherein a different output buffer is connected to a different one of the number of output ports. Each of the different output buffers associated with the number of output ports are connected to a switching substrate (e.g., crossbar switch or shared bus) which is also connected to a number of input ports. In this configuration, if two packets arrive from two different input ports at the same time and are destined for a common output port, only one packet can make progress. This situation exists because the two packets must share a connectivity into the common output port. Also in this configuration, a packet may be stored in any available output buffer, but the entire packet must be stored in the same output buffer. Therefore, the output buffer must have sufficient space to store the packet before the packet can progress.

[6] In either of the conventional packet buffer memory configurations mentioned above, a complicated buffer allocation mechanism is required to achieve efficient utilization of the input ports, output ports, and buffer memories. In order to avoid contention for resources when a new packet arrives, the system must take into account a

number of factors, among others, such as (a) availability of space in each buffer, (b) whether a packet is currently being deposited into or retrieved from each buffer, and (c) the number of packets that are currently waiting for deposition in each buffer.

[7] In view of the foregoing, there is a need for a more efficient shared buffering solution that can be implemented without requiring either a complicated buffer allocation scheme or high speed memory circuitry.

SUMMARY OF THE INVENTION

[8] Broadly speaking, the present invention fills these needs by providing an apparatus for efficiently utilizing a shared packet buffer memory in a switch and a method for operating the same. More specifically, the apparatus of the present invention includes a memory having a number of buffers configured to be operated in a ratcheted manner. The ratcheted manner in which the memory is operated causes each incoming data stream to be distributed across the number of buffers. Since each incoming data stream is stored in a distributed manner, a complicated buffer allocation scheme is not required. Each stored data stream can also be retrieved from the number of buffers for output from the memory in a similar ratcheted manner. The memory uses a common rotating selector to control the ratcheted manner of operation. Also, the memory is capable of simultaneously servicing each of a number of inputs and a number of outputs to which the memory is connected.

[9] In one embodiment, a switch is disclosed. The switch includes a set of inputs and a set of memory banks. A number of memory banks in the set of memory banks is equal to a number of inputs in the set of inputs. Each input is capable of transferring a data stream into the set of memory banks. The data stream transferred by each input is distributed across each of the set of memory banks. The switch further includes a set of outputs having a number outputs equal to the number of memory banks. Each data stream distributed across each of the set of memory banks is to be output from at least one output in the set of outputs.

[10] In another embodiment, a memory is disclosed. The memory includes a number of inputs that are each capable of receiving a data stream to be stored in the memory. The memory also includes memory banks equal in number to the inputs. The memory banks

are for storing the data streams received by the inputs. A first ratcheting distributor is provided for distributing the data stream received by either of the inputs across the memory banks. The ratcheting distributor distributes the data stream such that one or more of the memory banks contains a distinct portion of the data stream. The memory further includes a number of outputs that are each capable of providing the stream previously stored in the memory. The distinct portion of the data stream contained within either of the memory banks is provided to either of the number of outputs by a second ratcheting distributor.

[11] In another embodiment, a ratcheting distributor for distributing a data stream across a number of memory banks is disclosed. The ratcheting distributor includes a number of multiplexers that are each configured to receive a number of inputs and one of a number of selector signals. Each of the multiplexers has an output connected to one of the memory banks such that one of the multiplexers is connected to each of the memory banks. The selector signal received by each multiplexer is capable of controlling which of the number of inputs will be transmitted to the output. Each of the number of selector signals is defined by a different ordering of a number of individual signals. The number of individual signals is equal to the number of inputs. The ratcheting distributor also includes a rotating selector capable of generating the number of individual signals used to define the selector signals. On a given clock cycle, the rotating selector causes one of the individual signals to be a high signal and each of the remaining individual signals to be a low signal. The rotating selector further causes the high signal to be rotated among the number of individual signals on successive clock cycles. The ratcheting distributor also includes a clock for exercising the rotating selector.

[12] In another embodiment, a method for operating a memory is disclosed. One aspect of the method includes receiving a number of inputs that each represent a data stream to

be stored in the memory, wherein each of the number of inputs can be received simultaneously. The data stream associated with each of the inputs is distributed across a number of memory banks. Distribution of the data stream across the memory banks is performed such that a portion of the data stream is stored in each of the memory banks as required to completely store the data stream. Another aspect of the method includes retrieving the portion of the data stream stored in each of the memory banks. The portion of the data stream retrieved from storage in each of the memory banks is transmitted to an output such that the data stream is provided at the output.

[13] Other aspects of the invention will become more apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[14] The invention, together with further advantages thereof, may best be understood by reference to the following description taken in conjunction with the accompanying drawings in which:

Figure 1 is an illustration showing a memory, in accordance with one embodiment of the present invention;

Figure 2 is an illustration showing a rotating selector (i.e., ratchet), in accordance with one embodiment of the present invention;

Figure 3 is an illustration showing a table of selector signals corresponding to clock cycles 0 through 9, in accordance with the embodiment illustrated in Figures 1 and 2;

Figure 4A is an illustration showing the rotating selector at a clock cycle 0, in accordance with one embodiment of the present invention;

Figure 4B is an illustration showing the memory at the clock cycle 0, in accordance with one embodiment of the present invention;

Figure 5A is an illustration showing the rotating selector at a clock cycle 1, in accordance with one embodiment of the present invention;

Figure 5B is an illustration showing the memory at the clock cycle 1, in accordance with one embodiment of the present invention;

Figure 6A is an illustration showing the rotating selector at a clock cycle 2, in accordance with one embodiment of the present invention;

Figure 6B is an illustration showing the memory at the clock cycle 2, in accordance with one embodiment of the present invention;

Figure 7A is an illustration showing the rotating selector at a clock cycle 3, in accordance with one embodiment of the present invention;

Figure 7B is an illustration showing the memory at the clock cycle 3, in accordance with one embodiment of the present invention;

Figure 8A is an illustration showing the rotating selector at a clock cycle 4, in accordance with one embodiment of the present invention;

Figure 8B is an illustration showing the memory at the clock cycle 4, in accordance with one embodiment of the present invention;

Figure 9A is an illustration showing the rotating selector at a clock cycle 5, in accordance with one embodiment of the present invention;

Figure 9B is an illustration showing the memory at the clock cycle 5, in accordance with one embodiment of the present invention;

Figure 10A is an illustration showing the rotating selector at a clock cycle 6, in accordance with one embodiment of the present invention;

Figure 10B is an illustration showing the memory at the clock cycle 6, in accordance with one embodiment of the present invention;

Figure 11A is an illustration showing the rotating selector at a clock cycle 7, in accordance with one embodiment of the present invention;

Figure 11B is an illustration showing the memory at the clock cycle 7, in accordance with one embodiment of the present invention;

Figure 12A is an illustration showing the rotating selector at a clock cycle 8, in accordance with one embodiment of the present invention;

Figure 12B is an illustration showing the memory at the clock cycle 8, in accordance with one embodiment of the present invention;

Figure 13A is an illustration showing the rotating selector at a clock cycle 9, in accordance with one embodiment of the present invention;

Figure 13B is an illustration showing the memory at the clock cycle 9, in accordance with one embodiment of the present invention; and

Figure 14 shows a flowchart illustrating a method for operating a memory, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION

[15] Broadly speaking, the present invention provides an apparatus for efficiently utilizing a shared packet buffer memory in a switch and a method for operating the same. More specifically, the apparatus of the present invention includes a memory having a number of buffers configured to be operated in a ratcheted manner. The ratcheted manner in which the memory is operated causes each incoming data stream to be distributed across the number of buffers. Since each incoming data stream is stored in a distributed manner, a complicated buffer allocation scheme is not required. Each stored data stream can also be retrieved from the number of buffers for output from the memory in a similar ratcheted manner. The memory uses a rotating selector to control the ratcheted manner of operation. Also, the memory is capable of simultaneously servicing each of a number of inputs and a number of outputs to which the memory is connected. It should be appreciated that the present invention can be implemented in numerous ways, including as a process, an apparatus, a system, a device, or a method. Several exemplary embodiments of the invention will now be described in detail with reference to the accompanying drawings.

[16] In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

[17] Figure 1 is an illustration showing a memory 100, in accordance with one embodiment of the present invention. The memory 100 is capable of accepting four inputs I_A, I_B, I_C, and I_D and providing four outputs O_A, O_B, O_C, and O_D. The number of inputs

and outputs associated with the memory 100 are selected for exemplary purposes. In other embodiments, the memory 100 can include a different number of inputs and outputs. Each of the inputs I_A, I_B, I_C, and I_D are connected to four multiplexers (MUXs) 103A, 103B, 103C, and 103D. For purposes of illustration, a bus 101A is used to connect the four inputs I_A, I_B, I_C, and I_D to each of the MUXs 103A, 103B, 103C, and 103D. In actuality, the bus 101A can be considered to represent direct connections between each of the inputs I_A, I_B, I_C, and I_D and each of the MUXs 103A, 103B, 103C, and 103D. Each MUX 103A, 103B, 103C, and 103D receives a selector signal S_{W0}, S_{W1}, S_{W2}, and S_{W3}, respectively. Also, each MUX 103A, 103B, 103C, and 103D has an output W_0, W_1, W_2, and W_3, respectively, connected to a buffer (i.e., a memory bank) 105A, 105B, 105C, and 105D, respectively. In this manner, each output W_0, W_1, W_2, and W_3 serves as a write input to each buffer 105A, 105B, 105C, and 105D, respectively. Each selector signal S_{W0}, S_{W1}, S_{W2}, and S_{W3} is used to control which of the inputs I_A, I_B, I_C, and I_D will be transmitted to the output W_0, W_1, W_2, and W_3 of each of the MUXs 103A, 103B, 103C, and 103D, respectively. Thus, each selector signal S_{W0}, S_{W1}, S_{W2}, and S_{W3} controls which of the inputs I_A, I_B, I_C, and I_D will be written through the outputs W_0, W_1, W_2, and W_3 to each of the buffers 105A, 105B, 105C, and 105D, respectively.

[18] Each buffer 105A, 105B, 105C, and 105D contains a number of data storage locations 107 that are each capable of storing a word of data. Each buffer 105A, 105B, 105C, and 105D is also configured to communicate with an addressing controller 108. The addressing controller 108 provides each buffer with information related to which data storage location 107 is to be access in a particular input or output operation. Each buffer 105A, 105B, 105C, and 105D is also dual-ported to have the capability of simultaneously receiving input and providing output. In following, each buffer 105A, 105B, 105C, and 105D is connected to a read output R_0, R_1, R_2, and R_3, respectively. Each read output R_0,

R_1, R_2, and R_3 is connected to four MUXs 103E, 103F 103G, and 103H. For purposes of illustration, a bus 101B is used to connect each read output R_0, R_1, R_2, and R_3 to each of the MUXs 103E, 103F 103G, and 103H. In actuality, the bus 101B can be considered to represent direct connections between each read output R_0, R_1, R_2, and R_3 and each of the MUXs 103E, 103F 103G, and 103H. Each MUX 103E, 103F, 103G, and 103H receives a selector signal S_{R0}, S_{R1}, S_{R2}, and S_{R3}, respectively. Also, each MUX 103E, 103F, 103G, and 103H has an output O_A, O_B, O_C, and O_D, respectively, corresponding to the outputs of the memory 100. Each of the inputs I_A, I_B, I_C, and I_D, each of the write inputs W_0, W_1, W_2, and W_3, each of the read outputs R_0, R_1, R_2, and R_3, and each of the outputs O_A, O_B, O_C, and O_D have the same bandwidth capability.

[19] In the embodiment of Figure 1, each of the selector signals S_{W0}, S_{W1}, S_{W2}, S_{W3}, S_{R0}, S_{R1}, S_{R2}, and S_{R3} consists of four 1-bit signals with one of the 1-bit signals having a first digital state and the remainder of the 1-bit signals having a second digital state. In one embodiment, the first digital state is a high state and the second digital state is a low state. In another embodiment, the first digital state is a low state and the second digital state is a high state. For purposes of discussion and illustration, the remainder of this document will consider the first digital state to be a high state and the second digital state to be a low state. The number of 1-bit signals in the selector signal of a particular MUX is equal to the number of inputs received by the MUX. Since each of the MUXs 103A-103H is configured to receive four inputs, each selector signal contains four 1-bit signals. In other embodiments, the memory 100 may include a different number of inputs and buffers with a correspondingly different number of inputs to each MUX. In these other embodiments, the number of 1-bit signals in the selector signal of a particular MUX will continue to be equal to the number of inputs received by the MUX, with one 1-bit signal being high at a given time.

[20] The ordering of the 1-bit signals in the selector signals vary between the MUXs. Selector signals S_{W0} and S_{R0} have a 1-bit signal ordering of $\{S_3,S_2,S_1,S_0\}$ and correspond to MUXs 103A and 103E, respectively. Selector signals S_{W1} and S_{R1} have a 1-bit signal ordering of $\{S_0,S_3,S_2,S_1\}$ and correspond to MUXs 103B and 103F, respectively. Selector signals S_{W2} and S_{R2} have a 1-bit signal ordering of $\{S_1,S_0,S_3,S_2\}$ and correspond to MUXs 103C and 103G, respectively. Selector signals S_{W3} and S_{R3} have a 1-bit signal ordering of $\{S_2,S_1,S_0,S_3\}$ and correspond to MUXs 103D and 103H, respectively. Each of the selector signals maintains a common sequence of 1-bit signals. However, the common sequence of 1-bit signals is successively shifted by a bit position to create each of the selector signals. Thus, the high 1-bit signal will have a different bit position in each of the selector signals at a given time. The high 1-bit signal determines which of the inputs received by the MUX will be transmitted to the output of the MUX.

[21] With respect to MUX 103A, if the 1-bit signal S_0 is high in selector signal S_{W0}, input I_A will be transmitted to output W_0. If the 1-bit signal S_1 is high in selector signal S_{W0}, input I_B will be transmitted to output W_0. If the 1-bit signal S_2 is high in selector signal S_{W0}, input I_C will be transmitted to output W_0. If the 1-bit signal S_3 is high in selector signal S_{W0}, input I_D will be transmitted to output W_0.

[22] With respect to MUX 103B, if the 1-bit signal S_1 is high in selector signal S_{W1}, input I_A will be transmitted to output W_1. If the 1-bit signal S_2 is high in selector signal S_{W1}, input I_B will be transmitted to output W_1. If the 1-bit signal S_3 is high in selector signal S_{W1}, input I_C will be transmitted to output W_1. If the 1-bit signal S_0 is high in selector signal S_{W1}, input I_D will be transmitted to output W_1.

[23] With respect to MUX 103C, if the 1-bit signal S_2 is high in selector signal S_{W2}, input I_A will be transmitted to output W_2. If the 1-bit signal S_3 is high in selector signal S_{W2}, input I_B will be transmitted to output W_2. If the 1-bit signal S_0 is high in selector

signal S_{W2}, input I_C will be transmitted to output W_2. If the 1-bit signal S_1 is high in selector signal S_{W2}, input I_D will be transmitted to output W_2.

[24] With respect to MUX 103D, if the 1-bit signal S_3 is high in selector signal S_{W3}, input I_A will be transmitted to output W_3. If the 1-bit signal S_0 is high in selector signal S_{W3}, input I_B will be transmitted to output W_3. If the 1-bit signal S_1 is high in selector signal S_{W3}, input I_C will be transmitted to output W_3. If the 1-bit signal S_2 is high in selector signal S_{W3}, input I_D will be transmitted to output W_3.

[25] With respect to MUX 103E, if the 1-bit signal S_0 is high in selector signal S_{R0}, input R_0 will be transmitted to output O_A. If the 1-bit signal S_1 is high in selector signal S_{R0}, input R_1 will be transmitted to output O_A. If the 1-bit signal S_2 is high in selector signal S_{R0}, input R_2 will be transmitted to output O_A. If the 1-bit signal S_3 is high in selector signal S_{R0}, input R_3 will be transmitted to output O_A.

[26] With respect to MUX 103F, if the 1-bit signal S_1 is high in selector signal S_{R1}, input R_0 will be transmitted to output O_B. If the 1-bit signal S_2 is high in selector signal S_{R1}, input R_1 will be transmitted to output O_B. If the 1-bit signal S_3 is high in selector signal S_{R1}, input R_2 will be transmitted to output O_B. If the 1-bit signal S_0 is high in selector signal S_{R1}, input R_3 will be transmitted to output O_B.

[27] With respect to MUX 103G, if the 1-bit signal S_2 is high in selector signal S_{R2}, input R_0 will be transmitted to output O_C. If the 1-bit signal S_3 is high in selector signal S_{R2}, input R_1 will be transmitted to output O_C. If the 1-bit signal S_0 is high in selector signal S_{R2}, input R_2 will be transmitted to output O_C. If the 1-bit signal S_1 is high in selector signal S_{R2}, input R_3 will be transmitted to output O_C.

[28] With respect to MUX 103H, if the 1-bit signal S_3 is high in selector signal S_{R3}, input R_0 will be transmitted to output O_D. If the 1-bit signal S_0 is high in selector signal S_{R3}, input R_1 will be transmitted to output O_D. If the 1-bit signal S_1 is high in selector

signal S_{R3}, input R_2 will be transmitted to output O_D. If the 1-bit signal S_2 is high in selector signal S_{R3}, input R_3 will be transmitted to output O_D.

[29] Figure 2 is an illustration showing a rotating selector (i.e., ratchet) 200, in accordance with one embodiment of the present invention. The rotating selector 200 generates the 1-bit signals used to define each of the selector signals. The rotating selector 200 includes a number of cells 201A-201D corresponding to the number of 1-bit signals needed to generate each of the selector signals. Each cell 201A-201D receives a binary input (d) and a clock input and generates a binary output (q). The exemplary embodiment of Figure 2 corresponds to the memory 100 of Figure 1. Thus, four 1-bit signals S_0, S_1, S_2, and S_3 are generated by the cells 201A, 201B, 201C, and 201D, respectively. In one embodiment, each of the cells 201A-201D can be an edge-triggered cell such as a flip-flop. Upon receipt of a clock signal indicating a clock cycle, each of the cells 201A-201D will change the value of its internal state, and therefore its output (q), to take on the current value of its input. Thus, each of the cells 201A-201D transmits its state to the next cell in the sequence at each clock cycle.

[30] Each of the cells 201A-201D also includes a reset and set connection. Receipt of a high signal on the reset connection will cause the cell to hold a low signal. Receipt of a high signal on the set connection will cause the cell to hold a high signal. In the embodiment of Figure 2, a reset/initialize signal is connected to the set connection of cell 201A and to the reset connection of cells 201B-201D. Thus, in the embodiment of Figure 2, the cells 201A-201D can be initialized or reset such that cell 201A holds a high signal and each of cells 201B-201D holds a low signal. After initialization, the first clock cycle will cause cell 201B to generate a high signal and each of cells 201A, 201C, and 201D to generate a low signal. Thus, before the first clock cycle, S_0 is a high signal and S_1, S_2, and S_3 are low signals. In succeeding clock cycles, the high signal will rotate in sequence

from S_0 to S_1 to S_2 to S_3 and back to S_0 to start the rotation again. Therefore, one instance of the rotating selector 200 circuitry is needed to generate all of the 1-bit signals S_0, S_1, S_2, and S_3 needed to define each of the selector signals S_{W0}, S_{W1}, S_{W2}, S_{W3}, S_{R0}, S_{R1}, S_{R2}, and S_{R3}. Hence, each of the selector signals S_{W0}, S_{W1}, S_{W2}, S_{W3}, S_{R0}, S_{R1}, S_{R2}, and S_{R3} can be defined without using explicit arbitration circuitry.

[31] Figure 3 is an illustration showing a table of selector signals corresponding to clock cycles 0 through 9, in accordance with the embodiment illustrated in Figures 1 and 2. In clock cycle 0, the 1-bit signal S_0 is high and the remaining 1-bit signals S_1, S_2, and S_3 are low. In clock cycle 1, the 1-bit signal S_1 is high and the remaining 1-bit signals S_2, S_3, and S_0 are low. In clock cycle 2, the 1-bit signal S_2 is high and the remaining 1-bit signals S_3, S_0, and S_1 are low. In clock cycle 3, the 1-bit signal S_3 is high and the remaining 1-bit signals S_0, S_1, and S_2 are low. In clock cycle 4, the 1-bit signal S_0 is high again and the remaining 1-bit signals S_1, S_2, and S_3 are low again. The rotating of the high signal among the 1-bit signals continues in the same manner through the remaining clock cycles.

[32] The rotation of the high signal among the 1-bit signals at each clock cycle has a corresponding effect on the selection signals for each of the MUXs. At a given clock cycle, the high signal occupies a different bit position in each of the selector signals S_{W0}/S_{R0}, S_{W1}/S_{R1}, S_{W2}/S_{R2}, and S_{W3}/S_{R3}. In this manner, the selector signals S_{W0}, S_{W1}, S_{W2}, and S_{W3} control the MUXs 103A, 103B, 103C, and 103D, respectively, such that a data stream arriving at a given input will be distributed among the buffers 105A, 105B, 105C, and 105D as the clock cycles. In a similar manner, the selector signals S_{R0}, S_{R1}, S_{R2}, and S_{R3} control the MUXs 103E, 103F, 103G, and 103H, respectively, such that a data stream to be provided to a given output will be retrieved from the buffers 105A, 105B, 105C, and 105D as the clock cycles. Thus, the MUXs 103A, 103B, 103C, and

103D, along with the rotating selector 200, serve as a ratcheting distributor for distributing the data streams arriving at the various inputs I_A, I_B, I_C, and I_D among the buffers 105A, 105B, 105C, and 105D. Also, the MUXs 103E, 103F, 103G, and 103H, along with the rotating selector 200, serve as a ratcheting distributor for retrieving and transmitting the distributed data streams from among the buffers 105A, 105B, 105C, and 105D to the outputs O_A, O_B, O_C, and O_D. The operation of the memory 100 can be described more completely by way of example. Figures 4A-13B show an example of the rotating selector 200 and the memory 100 operation through ten clock cycles (cycle 0 through cycle 9).

[33] Figure 4A is an illustration showing the rotating selector 200 at a clock cycle 0, in accordance with one embodiment of the present invention. Figure 4B is an illustration showing the memory 100 at the clock cycle 0, in accordance with one embodiment of the present invention. Prior to the clock cycle 0 a number of data streams arrive at inputs I_A, I_B, I_C, and I_D. A memory controller receives requests from inputs I_A, I_B, I_C, and I_D to store the incoming data streams. The memory controller instructs the memory 100 to begin storing data streams incoming at inputs I_A, I_B, I_C, and I_D when S_{W0} is {0001}, {0010}, {0100}, and {1000}, respectively.

[34] At the clock cycle 0, S_{W0} is {0001}. Therefore, at the clock cycle 0, MUX 103A transmits a portion (i.e., a word) of the input I_A data stream through output W_0 to buffer 105A. Also, at or before the clock cycle 0 the addressing controller 108 establishes a data block across buffers 105A-105D for storage of the input I_A data stream. For illustrative purposes, the data block established for the input I_A data stream is identified by bracketed regions labeled I_A.0 at each buffer 105A-105D. The label I_A.0 refers to input I_A and data stream 0. In the current example, three data storage locations 107 are allocated within each buffer 105A-105D for storage of the input I_A data stream. The portion of the input I_A

data stream stored in buffer 105A at the clock cycle 0 is indicated by I_A.0.0, wherein the input identifier is I_A, the data stream identifier is 0, and the portion identifier is 0.

[35] In one embodiment, the addressing controller 108 will be provided information regarding the size of the incoming data stream to determine the number of data storage locations needed within each buffer to completely store the data stream. In one embodiment, the information regarding the size of the incoming data stream may be contained within a header of the data stream. In another embodiment, the information regarding the size of the incoming data stream may be discernable from the source of the incoming data stream.

[36] Figure 5A is an illustration showing the rotating selector 200 at a clock cycle 1, in accordance with one embodiment of the present invention. Figure 5B is an illustration showing the memory 100 at the clock cycle 1, in accordance with one embodiment of the present invention. At the clock cycle 1, S_{W0} is {0010}. Therefore, at the clock cycle 1, MUX 103A transmits a portion of the input I_B data stream through output W_0 to buffer 105A. Also, at or before the clock cycle 1 the addressing controller 108 establishes a data block across buffers 105A-105D for storage of the input I_B data stream. For illustrative purposes, the data block established for the input I_B data stream is identified by bracketed regions labeled I_B.0 at each buffer 105A-105D. The label I_B.0 refers to input I_B and data stream 0. In the current example, three data storage locations 107 are allocated within each buffer 105A-105D for storage of the input I_B data stream. The portion of the input I_B data stream stored in buffer 105A at the clock cycle 1 is indicated by I_B.0.0, wherein the input identifier is I_B, the data stream identifier is 0, and the portion identifier is 0.

[37] Also, at the clock cycle 1, the selector signal S_{W1} is {0001} which instructs MUX 103B to transmit a next portion of the input I_A data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of

the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in the buffer 105B. The next portion of the input I_A data stream stored in buffer 105B at the clock cycle 1 is indicated by I_A.0.1, wherein the portion identifier is 1.

[38] Figure 6A is an illustration showing the rotating selector 200 at a clock cycle 2, in accordance with one embodiment of the present invention. Figure 6B is an illustration showing the memory 100 at the clock cycle 2, in accordance with one embodiment of the present invention. At the clock cycle 2, S_{W0} is {0100}. Therefore, at the clock cycle 2, MUX 103A transmits a portion of the input I_C data stream through output W_0 to buffer 105A. Also, at or before the clock cycle 2 the addressing controller 108 establishes a data block across buffers 105A-105D for storage of the input I_C data stream. For illustrative purposes, the data block established for the input I_C data stream is identified by bracketed regions labeled I_C.0 at each buffer 105A-105D. The label I_C.0 refers to input I_C and data stream 0. In the current example, three data storage locations 107 are allocated within each buffer 105A-105D for storage of the input I_C data stream. The portion of the input I_C data stream stored in buffer 105A at the clock cycle 2 is indicated by I_C.0.0, wherein the input identifier is I_C, the data stream identifier is 0, and the portion identifier is 0.

[39] Also, at the clock cycle 2, the selector signal S_{W1} is {0010} which instructs MUX 103B to transmit a next portion of the input I_B data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105B. The next portion of the input I_B data stream stored in buffer 105B at the clock cycle 2 is indicated by I_B.0.1, wherein the portion identifier is 1.

[40] Also, at the clock cycle 2, the selector signal S_{W2} is {0001} which instructs MUX 103C to transmit a next portion of the input I_A data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of

the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in the buffer 105C. The next portion of the input I_A data stream stored in buffer 105C at the clock cycle 2 is indicated by I_A.0.2, wherein the portion identifier is 2.

[41] Figure 7A is an illustration showing the rotating selector 200 at a clock cycle 3, in accordance with one embodiment of the present invention. Figure 7B is an illustration showing the memory 100 at the clock cycle 3, in accordance with one embodiment of the present invention. At the clock cycle 3, S_{W0} is {1000}. Therefore, at the clock cycle 3, MUX 103A transmits a portion of the input I_D data stream through output W_0 to buffer 105A. Also, at or before the clock cycle 3 the addressing controller 108 establishes a data block across buffers 105A-105D for storage of the input I_D data stream. For illustrative purposes, the data block established for the input I_D data stream is identified by bracketed regions labeled I_D.0 at each buffer 105A-105D. The label I_D.0 refers to input I_D and data stream 0. Notice that the data block established for the input I_D data stream is not immediately adjacent to the data block established for the input I_C data stream. The addressing controller 108 can establish a data block for an incoming data stream at any available location in memory. In one embodiment, the addressing controller 108 is operated to establish data blocks for incoming data streams in a manner that will most efficiently utilize the available memory. In the current example, three data storage locations 107 are allocated within each buffer 105A-105D for storage of the input I_D data stream. The portion of the input I_D data stream stored in buffer 105A at the clock cycle 3 is indicated by I_D.0.0, wherein the input identifier is I_D, the data stream identifier is 0, and the portion identifier is 0.

[42] Also, at the clock cycle 3, the selector signal S_{W1} is {0100} which instructs MUX 103B to transmit a next portion of the input I_C data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of

the input I_C data stream at the next available data storage location 107 allocated for $I_C.0$ in the buffer 105B. The next portion of the input I_C data stream stored in buffer 105B at the clock cycle 3 is indicated by $I_C.0.1$, wherein the portion identifier is 1.

[43] Also, at the clock cycle 3, the selector signal S_{W2} is {0010} which instructs MUX 103C to transmit a next portion of the input I_B data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for $I_B.0$ in the buffer 105C. The next portion of the input I_B data stream stored in buffer 105C at the clock cycle 3 is indicated by $I_B.0.2$, wherein the portion identifier is 2.

[44] Also, at the clock cycle 3, the selector signal S_{W3} is {0001} which instructs MUX 103D to transmit a next portion of the input I_A data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for $I_A.0$ in the buffer 105D. The next portion of the input I_A data stream stored in buffer 105D at the clock cycle 3 is indicated by $I_A.0.3$, wherein the portion identifier is 3.

[45] Figure 8A is an illustration showing the rotating selector 200 at a clock cycle 4, in accordance with one embodiment of the present invention. Figure 8B is an illustration showing the memory 100 at the clock cycle 4, in accordance with one embodiment of the present invention. At the clock cycle 4, the selector signal S_{W0} is {0001} which instructs MUX 103A to transmit a next portion of the input I_A data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for $I_A.0$ in the buffer 105A. The next portion of the input I_A data stream stored in buffer 105A at the clock cycle 4 is indicated by $I_A.0.4$, wherein the portion identifier is 4.

[46] Also, at the clock cycle 4, the selector signal S_{W1} is {1000} which instructs MUX 103B to transmit a next portion of the input I_D data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for I_D.0 in the buffer 105B. The next portion of the input I_D data stream stored in buffer 105B at the clock cycle 4 is indicated by I_D.0.1, wherein the portion identifier is 1.

[47] Also, at the clock cycle 4, the selector signal S_{W2} is {0100} which instructs MUX 103C to transmit a next portion of the input I_C data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for I_C.0 in the buffer 105C. The next portion of the input I_C data stream stored in buffer 105C at the clock cycle 4 is indicated by I_C.0.2, wherein the portion identifier is 2.

[48] Also, at the clock cycle 4, the selector signal S_{W3} is {0010} which instructs MUX 103D to transmit a next portion of the input I_B data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105D. The next portion of the input I_B data stream stored in buffer 105D at the clock cycle 4 is indicated by I_B.0.3, wherein the portion identifier is 3.

[49] Figure 9A is an illustration showing the rotating selector 200 at a clock cycle 5, in accordance with one embodiment of the present invention. Figure 9B is an illustration showing the memory 100 at the clock cycle 5, in accordance with one embodiment of the present invention. At the clock cycle 5, the selector signal S_{W0} is {0010} which instructs MUX 103A to transmit a next portion of the input I_B data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_B data stream at the next available data storage location 107 allocated

for $I_B.0$ in the buffer 105A. The next portion of the input I_B data stream stored in buffer 105A at the clock cycle 5 is indicated by $I_B.0.4$, wherein the portion identifier is 4.

[50] Also, at the clock cycle 5, the selector signal S_{W1} is {0001} which instructs MUX 103B to transmit a next portion of the input I_A data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for $I_A.0$ in the buffer 105B. The next portion of the input I_A data stream stored in buffer 105B at the clock cycle 5 is indicated by $I_A.0.5$, wherein the portion identifier is 5.

[51] Also, at the clock cycle 5, the selector signal S_{W2} is {1000} which instructs MUX 103C to transmit a next portion of the input I_D data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for $I_D.0$ in the buffer 105C. The next portion of the input I_D data stream stored in buffer 105C at the clock cycle 5 is indicated by $I_D.0.2$, wherein the portion identifier is 2.

[52] Also, at the clock cycle 5, the selector signal S_{W3} is {0100} which instructs MUX 103D to transmit a next portion of the input I_C data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for $I_C.0$ in the buffer 105D. The next portion of the input I_C data stream stored in buffer 105D at the clock cycle 5 is indicated by $I_C.0.3$, wherein the portion identifier is 3.

[53] Continuing with the current example, prior to the clock cycle 5, the memory controller receives requests to provide the stored data streams $I_C.0$, $I_D.0$, $I_A.0$, and $I_B.0$ to outputs O_A, O_B, O_C, and O_D, respectively. The memory controller instructs the memory 100 to begin providing the stored data streams to outputs O_A, O_B, O_C, and O_D when S_{R0}, S_{R1}, S_{R2}, and S_{R3}, respectively, are {0001}. At the clock cycle 5, S_{R1} is {0001}.

Therefore, at the clock cycle 5, MUX 103F transmits the I_D.0.0 portion of the input I_D data stream from the buffer 105A output R_0 to output O_B.

[54] Figure 10A is an illustration showing the rotating selector 200 at a clock cycle 6, in accordance with one embodiment of the present invention. Figure 10B is an illustration showing the memory 100 at the clock cycle 6, in accordance with one embodiment of the present invention. At the clock cycle 6, the selector signal S_{W0} is {0100} which instructs MUX 103A to transmit a next portion of the input I_C data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for I_C.0 in the buffer 105A. The next portion of the input I_C data stream stored in buffer 105A at the clock cycle 6 is indicated by I_C.0.4, wherein the portion identifier is 4.

[55] Also, at the clock cycle 6, the selector signal S_{W1} is {0010} which instructs MUX 103B to transmit a next portion of the input I_B data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105B. The next portion of the input I_B data stream stored in buffer 105B at the clock cycle 6 is indicated by I_B.0.5, wherein the portion identifier is 5.

[56] Also, at the clock cycle 6, the selector signal S_{W2} is {0001} which instructs MUX 103C to transmit a next portion of the input I_A data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in the buffer 105C. The next portion of the input I_A data stream stored in buffer 105C at the clock cycle 6 is indicated by I_A.0.6, wherein the portion identifier is 6.

[57] Also, at the clock cycle 6, the selector signal S_{W3} is {1000} which instructs MUX 103D to transmit a next portion of the input I_D data stream through output W_3 to buffer

105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for I_D.0 in the buffer 105D. The next portion of the input I_D data stream stored in buffer 105D at the clock cycle 6 is indicated by I_D.0.3, wherein the portion identifier is 3.

[58] Also, at the clock cycle 6, S_{R2} is {0001}. Therefore, at the clock cycle 6, MUX 103G transmits the I_A.0.0 portion of the input I_A data stream from the buffer 105A output R_0 to output O_C.

[59] Also, at the clock cycle 6, S_{R1} is {0010}. Therefore, at the clock cycle 6, MUX 103G transmits the I_D.0.1 portion of the input I_D data stream from the buffer 105B output R_1 to output O_B.

[60] Figure 11A is an illustration showing the rotating selector 200 at a clock cycle 7, in accordance with one embodiment of the present invention. Figure 11B is an illustration showing the memory 100 at the clock cycle 7, in accordance with one embodiment of the present invention. At the clock cycle 7, the selector signal S_{W0} is {1000} which instructs MUX 103A to transmit a next portion of the input I_D data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for I_D.0 in the buffer 105A. The next portion of the input I_D data stream stored in buffer 105A at the clock cycle 7 is indicated by I_D.0.4, wherein the portion identifier is 4.

[61] Also, at the clock cycle 7, the selector signal S_{W1} is {0100} which instructs MUX 103B to transmit a next portion of the input I_C data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for I_C.0 in the buffer 105B. The next portion of the input I_C data stream stored in buffer 105B at the clock cycle 7 is indicated by I_C.0.5, wherein the portion identifier is 5.

[62] Also, at the clock cycle 7, the selector signal S_{W2} is {0010} which instructs MUX 103C to transmit a next portion of the input I_B data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105C. The next portion of the input I_B data stream stored in buffer 105C at the clock cycle 7 is indicated by I_B.0.6, wherein the portion identifier is 6.

[63] Also, at the clock cycle 7, the selector signal S_{W3} is {0001} which instructs MUX 103D to transmit a next portion of the input I_A data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in the buffer 105D. The next portion of the input I_A data stream stored in buffer 105D at the clock cycle 7 is indicated by I_A.0.7, wherein the portion identifier is 7.

[64] Also, at the clock cycle 7, S_{R3} is {0001}. Therefore, at the clock cycle 7, MUX 103H transmits the I_B.0.0 portion of the input I_B data stream from the buffer 105A output R_0 to output O_D.

[65] Also, at the clock cycle 7, S_{R2} is {0010}. Therefore, at the clock cycle 7, MUX 103G transmits the I_A.0.1 portion of the input I_A data stream from the buffer 105B output R_1 to output O_C.

[66] Also, at the clock cycle 7, S_{R1} is {0100}. Therefore, at the clock cycle 7, MUX 103F transmits the I_D.0.2 portion of the input I_D data stream from the buffer 105C output R_2 to output O_B.

[67] Figure 12A is an illustration showing the rotating selector 200 at a clock cycle 8, in accordance with one embodiment of the present invention. Figure 12B is an illustration showing the memory 100 at the clock cycle 8, in accordance with one embodiment of the present invention. At the clock cycle 8, the selector signal S_{W0} is {0001} which instructs

MUX 103A to transmit a next portion of the input I_A data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in the buffer 105A. The next portion of the input I_A data stream stored in buffer 105A at the clock cycle 8 is indicated by I_A.0.8, wherein the portion identifier is 8.

[68] Also, at the clock cycle 8, the selector signal S_{W1} is {1000} which instructs MUX 103B to transmit a next portion of the input I_D data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for I_D.0 in the buffer 105B. The next portion of the input I_D data stream stored in buffer 105B at the clock cycle 8 is indicated by I_D.0.5, wherein the portion identifier is 5.

[69] Also, at the clock cycle 8, the selector signal S_{W2} is {0100} which instructs MUX 103C to transmit a next portion of the input I_C data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for I_C.0 in the buffer 105C. The next portion of the input I_C data stream stored in buffer 105C at the clock cycle 8 is indicated by I_C.0.6, wherein the portion identifier is 6.

[70] Also, at the clock cycle 8, the selector signal S_{W3} is {0010} which instructs MUX 103D to transmit a next portion of the input I_B data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105D. The next portion of the input I_B data stream stored in buffer 105D at the clock cycle 8 is indicated by I_B.0.7, wherein the portion identifier is 7.

[71] Also, at the clock cycle 8, S_{R0} is {0001}. Therefore, at the clock cycle 8, MUX 103E transmits the I_C.0.0 portion of the input I_C data stream from the buffer 105A output R_0 to output O_A.

[72] Also, at the clock cycle 8, S_{R3} is {0010}. Therefore, at the clock cycle 8, MUX 103H transmits the I_B.0.1 portion of the input I_B data stream from the buffer 105B output R_1 to output O_D.

[73] Also, at the clock cycle 8, S_{R2} is {0100}. Therefore, at the clock cycle 8, MUX 103G transmits the I_A.0.2 portion of the input I_A data stream from the buffer 105C output R_2 to output O_C.

[74] Also, at the clock cycle 8, S_{R1} is {1000}. Therefore, at the clock cycle 8, MUX 103F transmits the I_D.0.3 portion of the input I_D data stream from the buffer 105D output R_3 to output O_B.

[75] Figure 13A is an illustration showing the rotating selector 200 at a clock cycle 9, in accordance with one embodiment of the present invention. Figure 13B is an illustration showing the memory 100 at the clock cycle 9, in accordance with one embodiment of the present invention. At the clock cycle 9, the selector signal S_{W0} is {0010} which instructs MUX 103A to transmit a next portion of the input I_B data stream through output W_0 to buffer 105A. The addressing controller 108 directs the buffer 105A to store the next portion of the input I_B data stream at the next available data storage location 107 allocated for I_B.0 in the buffer 105A. The next portion of the input I_B data stream stored in buffer 105A at the clock cycle 9 is indicated by I_B.0.8, wherein the portion identifier is 8.

[76] Also, at the clock cycle 9, the selector signal S_{W1} is {0001} which instructs MUX 103B to transmit a next portion of the input I_A data stream through output W_1 to buffer 105B. The addressing controller 108 directs the buffer 105B to store the next portion of the input I_A data stream at the next available data storage location 107 allocated for I_A.0 in

the buffer 105B. The next portion of the input I_A data stream stored in buffer 105B at the clock cycle 9 is indicated by I_A.0.9, wherein the portion identifier is 9.

[77] Also, at the clock cycle 9, the selector signal S_{W2} is {1000} which instructs MUX 103C to transmit a next portion of the input I_D data stream through output W_2 to buffer 105C. The addressing controller 108 directs the buffer 105C to store the next portion of the input I_D data stream at the next available data storage location 107 allocated for I_D.0 in the buffer 105C. The next portion of the input I_D data stream stored in buffer 105C at the clock cycle 9 is indicated by I_D.0.6, wherein the portion identifier is 6.

[78] Also, at the clock cycle 9, the selector signal S_{W3} is {0100} which instructs MUX 103D to transmit a next portion of the input I_C data stream through output W_3 to buffer 105D. The addressing controller 108 directs the buffer 105D to store the next portion of the input I_C data stream at the next available data storage location 107 allocated for I_C.0 in the buffer 105D. The next portion of the input I_C data stream stored in buffer 105D at the clock cycle 9 is indicated by I_C.0.7, wherein the portion identifier is 7.

[79] Also, at the clock cycle 9, S_{R0} is {0010}. Therefore, at the clock cycle 9, MUX 103E transmits the I_C.0.1 portion of the input I_C data stream from the buffer 105B output R_1 to output O_A.

[80] Also, at the clock cycle 9, S_{R3} is {0100}. Therefore, at the clock cycle 9, MUX 103H transmits the I_B.0.2 portion of the input I_B data stream from the buffer 105C output R_2 to output O_D.

[81] Also, at the clock cycle 9, S_{R2} is {1000}. Therefore, at the clock cycle 9, MUX 103G transmits the I_A.0.3 portion of the input I_A data stream from the buffer 105D output R_3 to output O_C.

[82] Also, at the clock cycle 9, S_{R1} is {0001}. Therefore, at the clock cycle 9, MUX 103F transmits the I_D.0.4 portion of the input I_D data stream from the buffer 105A output R_0 to output O_B.

[83] In this example S_{R0}-S_{R3} are composed of the same signals that make up S_{W0}-S_{W3}. However, they may also be generated by a distinct rotating selector.

[84] As demonstrated by the example memory 100 and rotating selector 200 operation shown in Figures 4A-13B, when a data stream is deposited in the memory 100, the data stream is interleaved over a number of memory banks included in the memory 100. Each of the number of memory banks is independently addressable. In one embodiment, each of the number of memory banks is also dual ported to be written to and read from simultaneously. However, in other embodiments, memory banks that are not dual ported can also be used. The memory 100 is capable of simultaneously receiving a number of incoming data stream transmissions and providing a number of outgoing data stream transmissions. Each of the incoming and outgoing data stream transmissions can be independently started and stopped by a memory controller. In one embodiment, once an incoming or outgoing data stream transmission is started, the data stream transmission tends to continue for a while before an idle period is encountered. The tendency of the data stream transmission to continue is not a necessary condition of the present invention, but is a characteristic of switch traffic that can be exploited by the present invention for better performance.

[85] As previously discussed, the memory controller instructs the memory 100 to begin storing data streams incoming at inputs I_A, I_B, I_C, and I_D when S_{W0} is {0001}, {0010}, {0100}, and {1000}, respectively. If storage of the incoming data streams cannot be started or stopped immediately (i.e., if a latency exists from the time a start or stop instruction is issued by the memory controller to the time the incoming data stream

actually starts or stops), an elasticity buffer can be added before each of the inputs I_A, I_B, I_C, and I_D. The elasticity buffer can be used to absorb data that continues to arrive after a stop instruction is issued by the memory controller, and to buffer up data that arrives prior to execution of a start instruction issued by the memory controller. In the example memory 100, each of the inputs I_A, I_B, I_C, and I_D will have to wait at most four clock cycles before execution of a start instruction issued by the memory controller. Therefore, the elasticity buffer should be large enough to handle an amount of data that can arrive in an incoming data stream over four clock cycles.

[86] Since each memory bank is assigned to a different input and output in every clock cycle, the addressing controller 108 changes the write and read addresses for each memory bank in every clock cycle to point to the appropriate data storage location. In one embodiment, management of the addresses for each memory bank by the addressing controller 108 can be performed using a time slicing device similar to that described in the patent application entitled "Time Slicing Device for Shared Resources and Method for Operating the Same," as previously incorporated herein by reference.

[87] The memory 100 as previously described assumes that all of the inputs and all of the outputs have the same bandwidth characteristics. In applications where there exist inputs of differing bandwidth characteristics, a bandwidth matching device similar to that described in the patent application entitled "Bandwidth Matching Device and Method for Operating the Same," as previously incorporated herein by reference, can be used. The bandwidth matching device is capable of transforming the input bandwidths to satisfy a common bandwidth requirement before being sent to the ratcheting distributor. For example, if there are four 1x bandwidth inputs and three 4x bandwidth inputs, the four 1x inputs can be merged into a single 4x stream using the bandwidth matching device. Then, the resulting four 4x streams can be sent to the ratcheting distributor. Alternatively, the

three 4x bandwidth inputs can be transformed into twelve 1x streams. Then, the resulting sixteen 1x streams can be sent to the ratcheting distributor.

[88] Figure 14 shows a flowchart illustrating a method for operating a memory, in accordance with one embodiment of the present invention. The method includes an operation 1401 in which the memory receives a number of inputs that each represent a data stream to be stored in the memory. The memory is operated with a capability to receive each of the number of inputs simultaneously. The method also includes an operation 1403 in which the data stream associated with each of the number of inputs is distributed across a number of memory banks. A portion of the data stream is stored in each of the number of memory banks as required to completely store the data stream.

[89] A ratcheting distributor can be used to distribute the data stream associated with each of the number of inputs across the number of memory banks. Use of the ratcheting distributor includes operation of a number of multiplexers. When operated, each of the number of multiplexers receive each of the number of inputs and a selector signal. The selector signal controls which of the number of inputs is transmitted to a multiplexer output that is connected to one of the number of memory banks. Use of the ratcheting distributor further includes generating the selector signal received by each of the number of multiplexers. The generated selector signal is different for each of the number of multiplexers on each cycle of a clock. However, the number of different selector signals is equal to the number of multiplexers. The selector signal received by each of the number of multiplexers is repeated on a clock cycle that is a multiple of the number of inputs.

[90] The method further includes an operation 1405 in which the portion of the data stream stored in each of the number of memory banks is retrieved. In an operation 1407, the retrieved portions of the data stream are transmitted to an output. Transmission of the retrieved portions of the data stream to the output cause the data stream to be provided at

the output. The memory is operated with a capability to simultaneously retrieve and transmit portions of different data streams stored in each of the number of memory banks, such that each of the number of data streams are transmitted to different outputs.

[91] A second ratcheting distributor can be used to retrieve and transmit the portion of the data stream stored in each of the number of memory banks. Use of the second ratcheting distributor includes operation of a number of multiplexers. When operated, each of the number of multiplexers receive the portion of the data stream stored in each of the number of memory banks and a selector signal. The selector signal controls which portion of the data stream stored in each of the number of memory banks is transmitted to the output. Use of the ratcheting distributor further includes generating the selector signal received by each of the number of multiplexers. The generated selector signal is different for each of the number of multiplexers on each cycle of a clock. However, the number of different selector signals is equal to the number of multiplexers. The selector signal received by each of the number of multiplexers is repeated on a clock cycle that is a multiple of the number of memory banks.

[92] While this invention has been described in terms of several embodiments, it will be appreciated that those skilled in the art upon reading the preceding specifications and studying the drawings will realize various alterations, additions, permutations and equivalents thereof. It is therefore intended that the present invention includes all such alterations, additions, permutations, and equivalents as fall within the true spirit and scope of the invention.

What is claimed is: